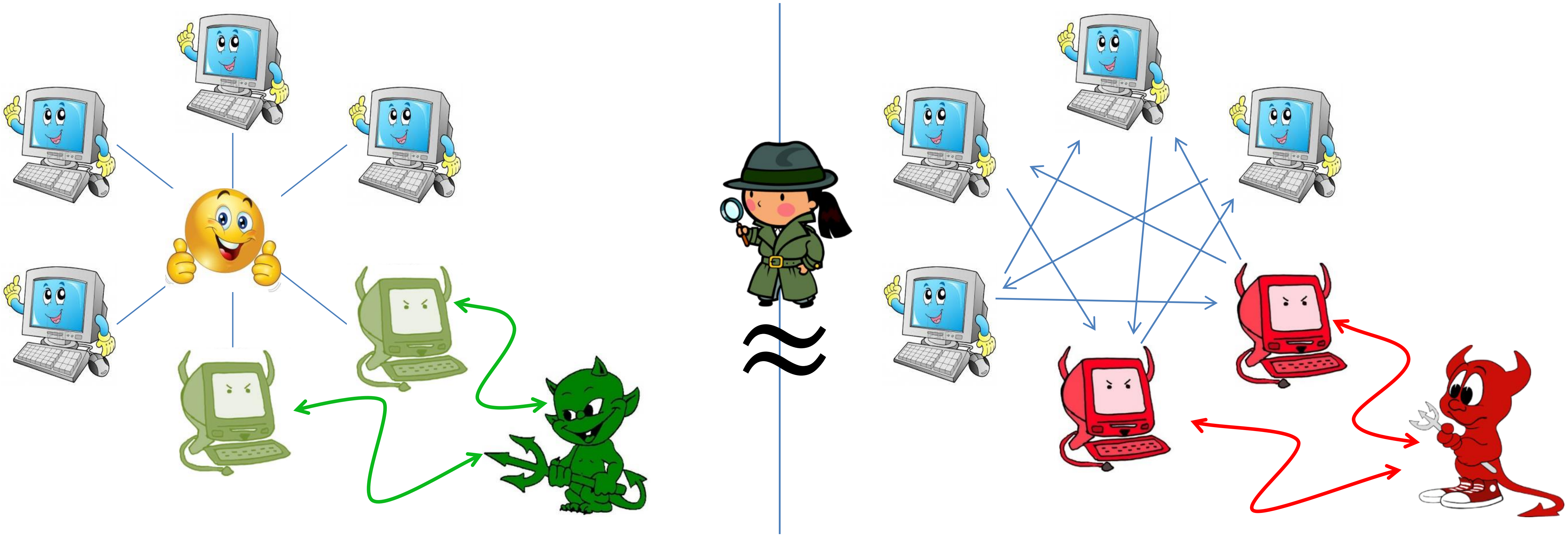
Completeness Theorems for Adaptively Secure Broadcast

Ran Cohen Juan Garay Vassilis Zikas

Reichman University

Secure Multiparty Computation (MPC)

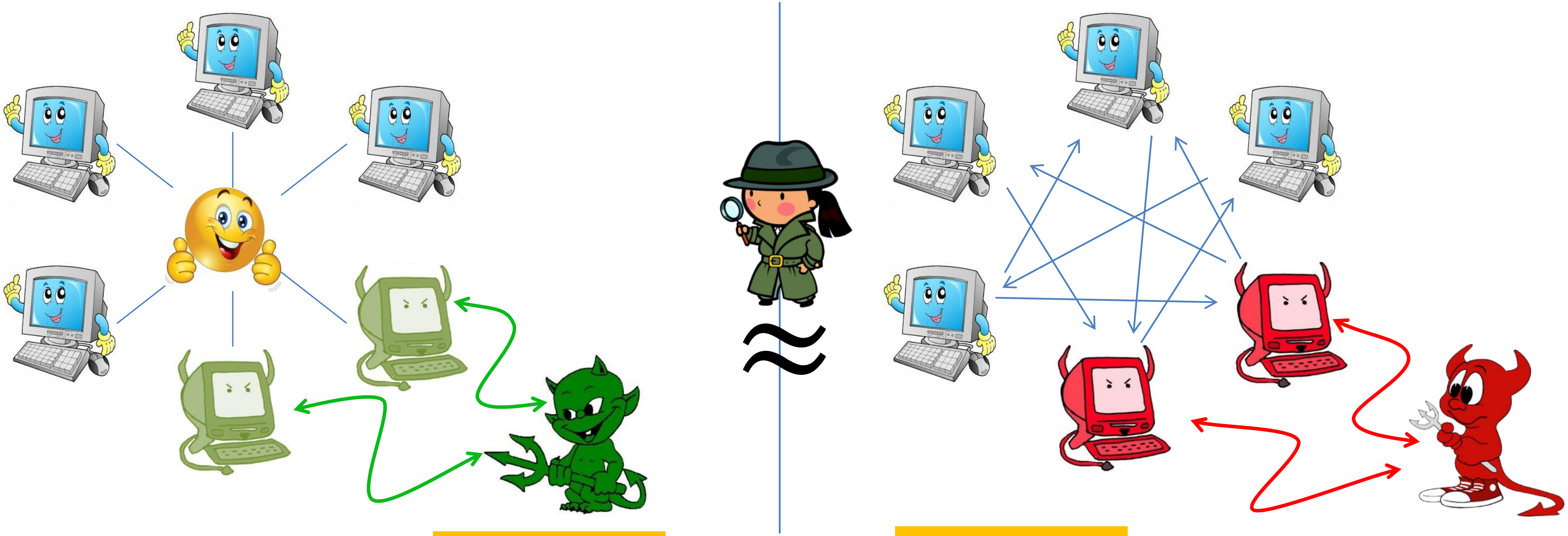
Jointly compute on secret data, **without revealing the data**



A protocol is secure if ∀ real-world adversary ∃ ideal-world adversary such that no environment can distinguish real from ideal

Secure Multiparty Computation (MPC)

Adaptive corruptions?



adaptive

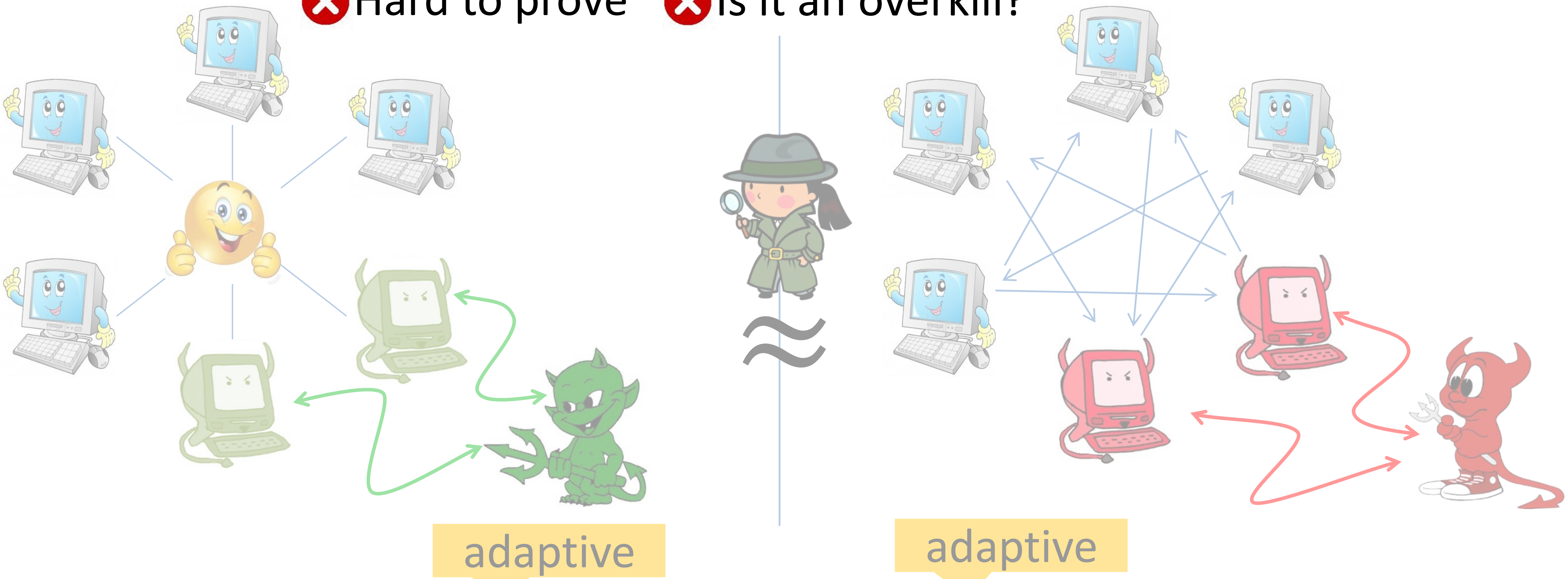
adaptive

A protocol is secure if ∀ real-world adversary ∃ ideal-world adversary such that no environment can distinguish real from ideal

Secure Multiparty Computation (MPC)

✓ Holistic definition ✓ Composition ✓ Clear meaning of security

✗ Hard to prove ✗ Is it an overkill?

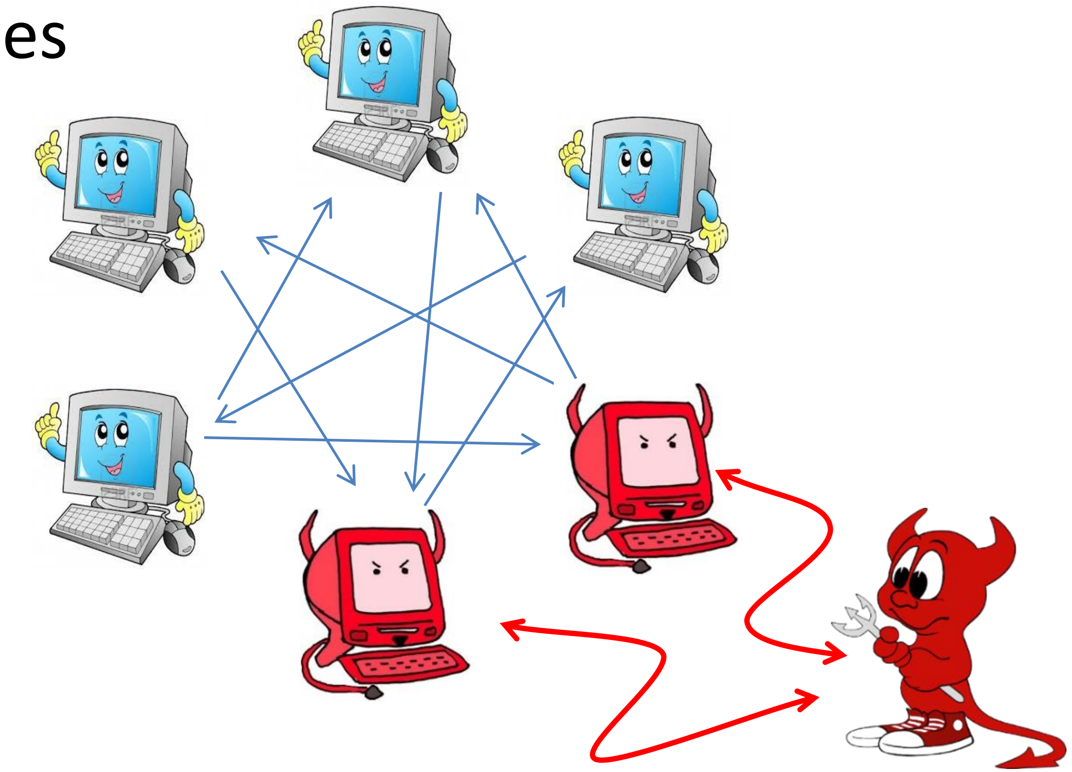


A protocol is secure if ∀ real-world adversary ∃ ideal-world adversary such that no environment can distinguish real from ideal

MPC: Property based

A protocol is secure if the following properties are satisfied against any XYX adversary:

- Correctness
- Privacy
- Independence of inputs
- Fairness
- Guaranteed output delivery



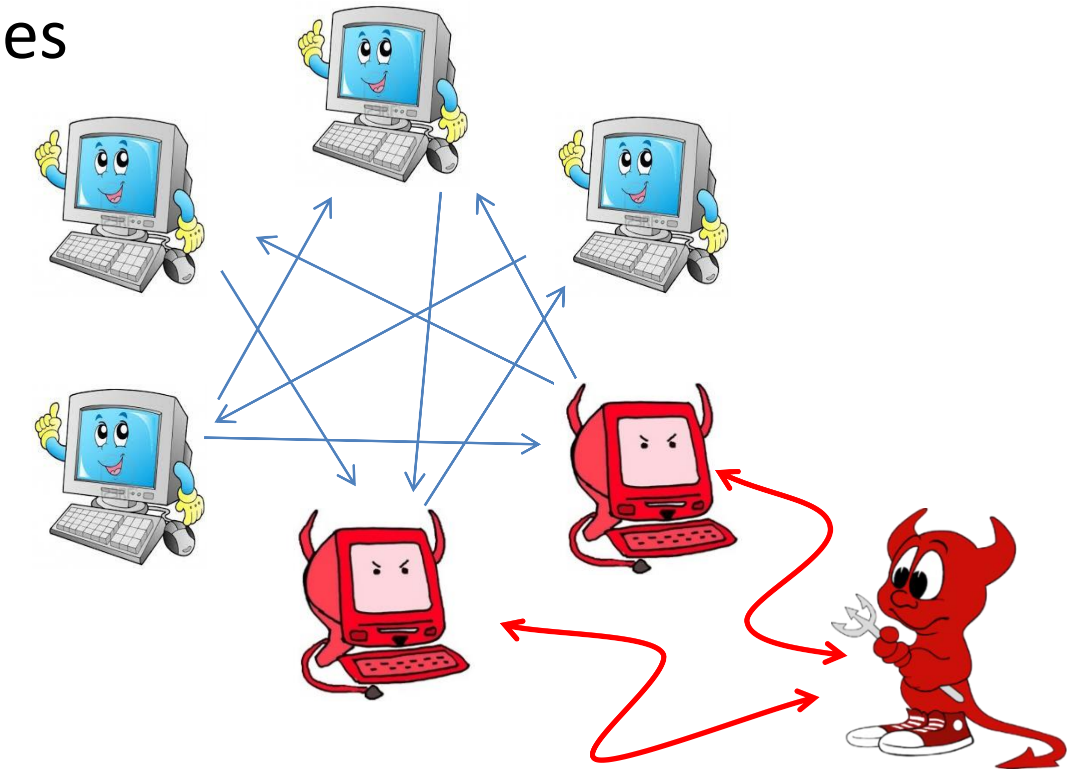
MPC: Property based

Adaptive corruptions?

A protocol is secure if the following properties are satisfied against any XYX adversary:

adaptive

- Correctness
- Privacy
- Independence of inputs
- Fairness
- Guaranteed output delivery



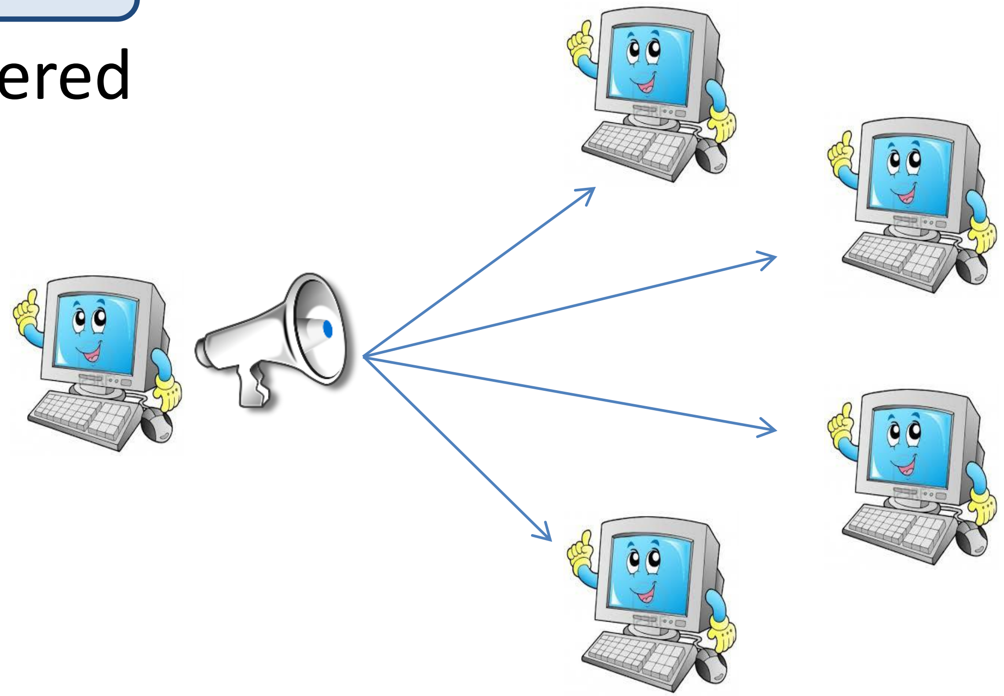
Should a protocol satisfying those properties in the presence of an **adaptive adversary** be considered **adaptively secure**?

Case study: Adaptively Secure Broadcast

Goal: emulate a broadcast channel

A broadcast protocol with sender S is considered secure if it satisfies the following properties:

- Agreement: every honest party outputs the same value y
- Validity: if the sender is honest and has input x, then $y = x$



Should a broadcast protocol satisfying those properties in the presence of an **adaptive adversary** be considered **adaptively secure**?

NOOOO!!!

Case study: Adaptively Secure Broadcast

Goal: emulate a broadcast channel

A broadcast protocol with sender S is considered secure if it satisfies the following properties:

- Agreement: every honest party outputs the same value y
- Validity: if the sender is honest and has input x, then $y = x$

at the end of the protocol

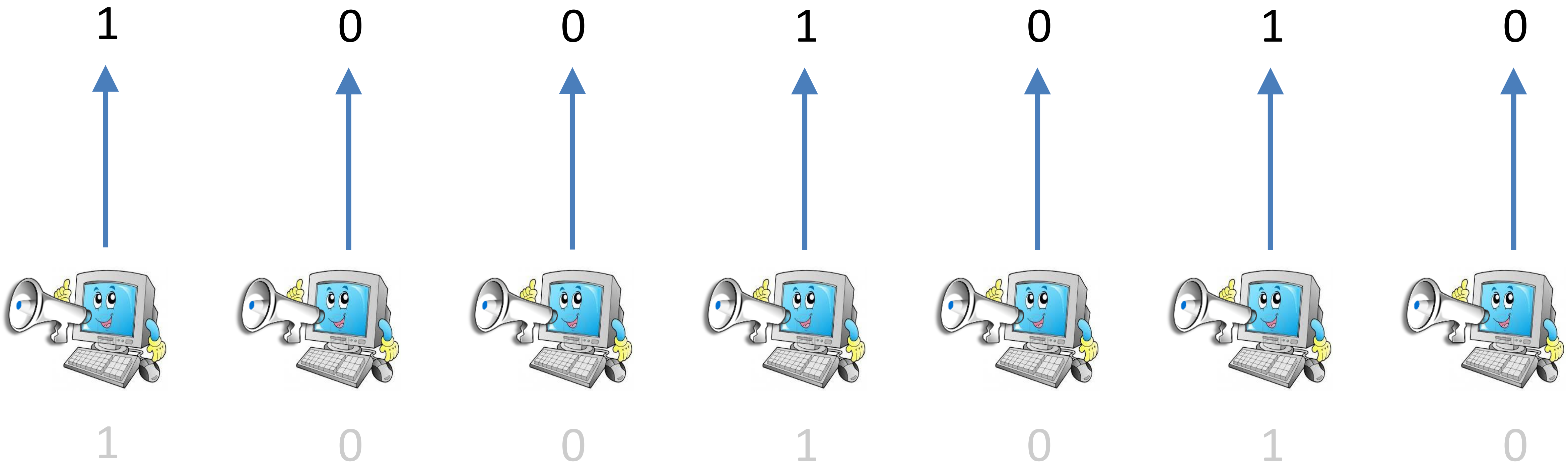
until the end of the protocol

Should a broadcast protocol satisfying those properties in the presence of an **adaptive adversary** be considered **adaptively secure**?

MMAYBE??

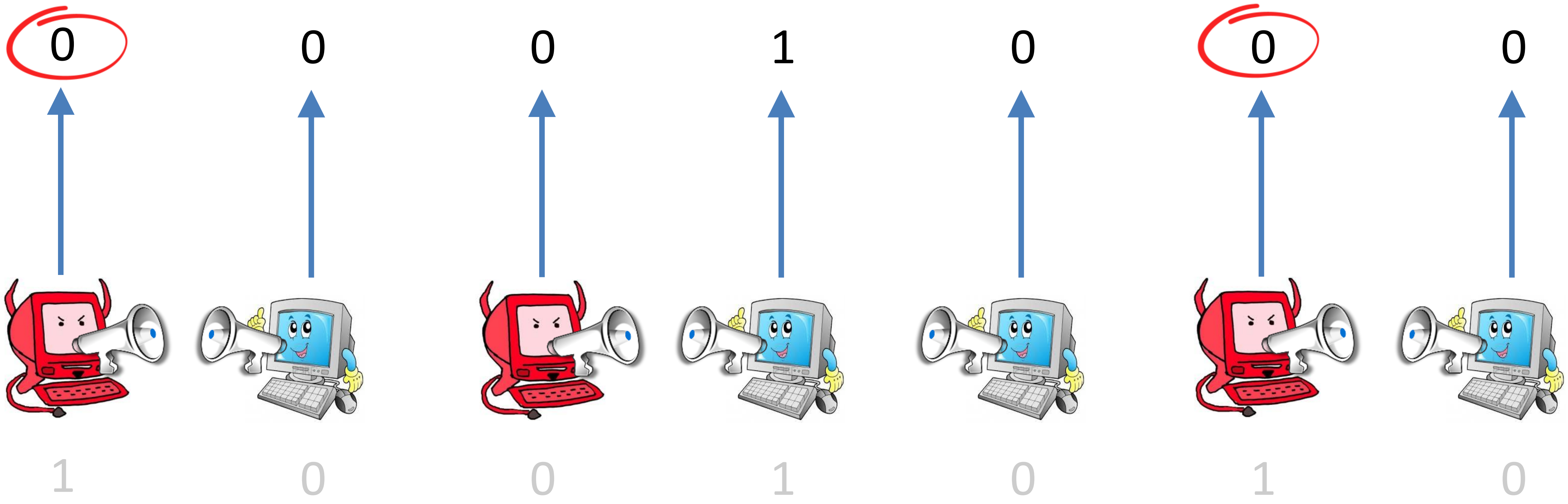
Case study: Adaptively Secure Broadcast

Problem: everybody broadcasts a bit; the adversary wants the output to be (as close as possible to) 0000…000



Case study: Adaptively Secure Broadcast

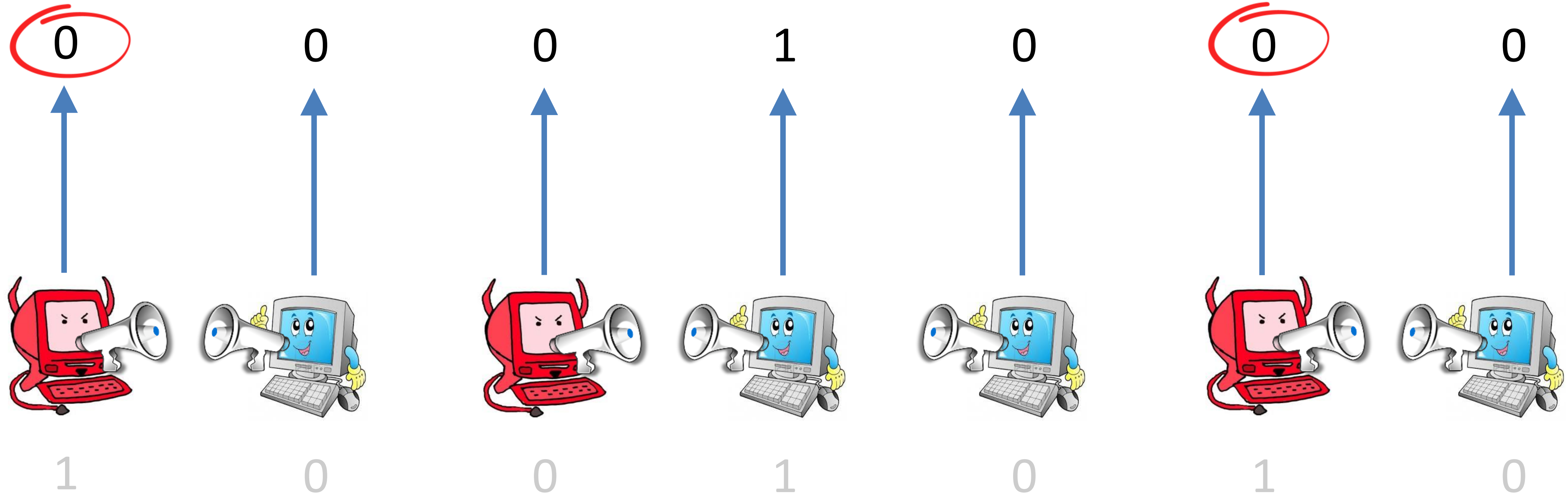
Problem: everybody broadcasts a bit; the adversary wants the output to be (as close as possible to) 0000…000



Static adversary, 3 corruptions

Case study: Adaptively Secure Broadcast

Problem: everybody broadcasts a bit; the adversary wants the output to be (as close as possible to) 0000…000

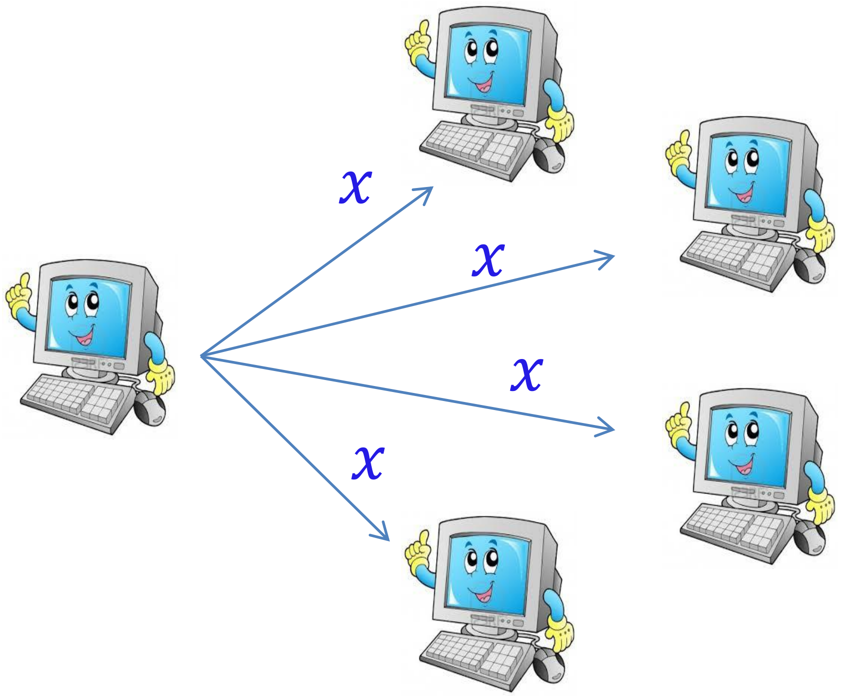


Adaptive adversary, 3 corruptions, broadcast channel

What if we use a broadcast protocol?

(Almost) all known broadcast protocols follow this paradigm:

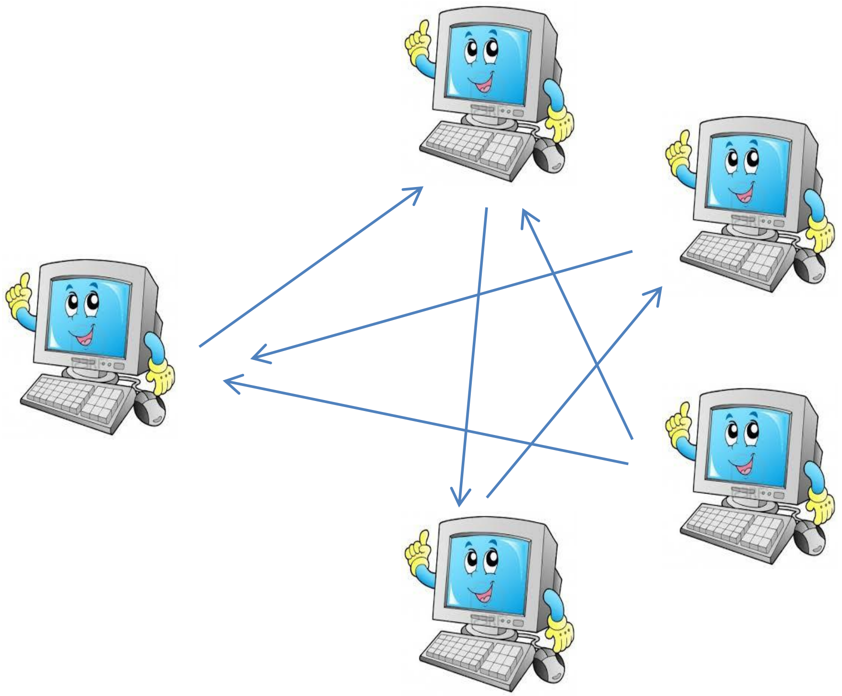
- Step 1: Sender sends its input x to every party
- Step 2: Parties try to establish agreement



What if we use a broadcast protocol?

(Almost) all known broadcast protocols follow this paradigm:

- Step 1: Sender sends its input x to every party
- Step 2: Parties try to establish agreement

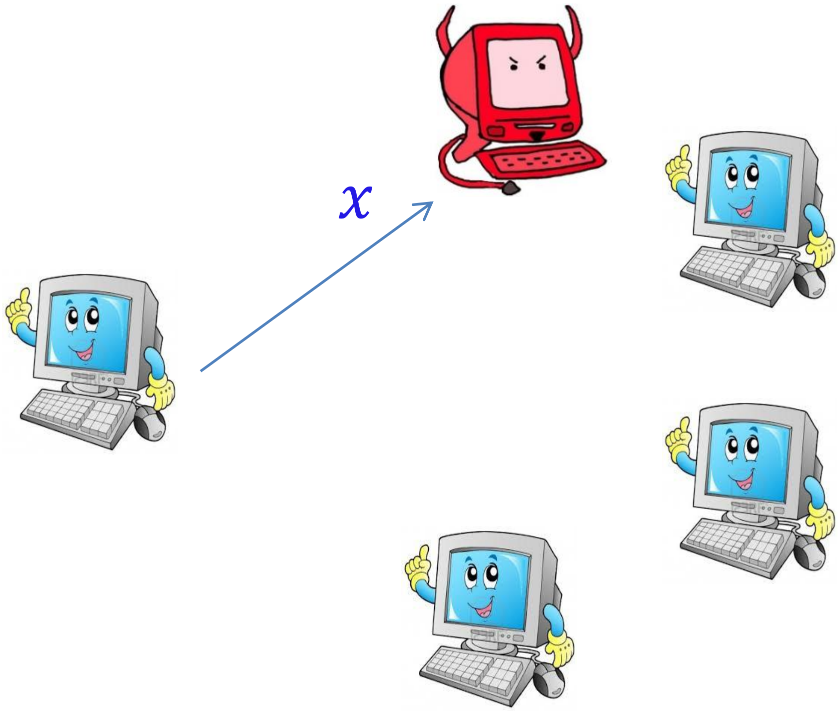


What if we use a broadcast protocol?

(Almost) all known broadcast protocols follow this paradigm:

- Step 1: Sender sends its input x to every party
- Step 2: Parties try to establish agreement

All these protocols satisfy agreement and validity, even facing an adaptive adversary
Should they be considered adaptively secure?



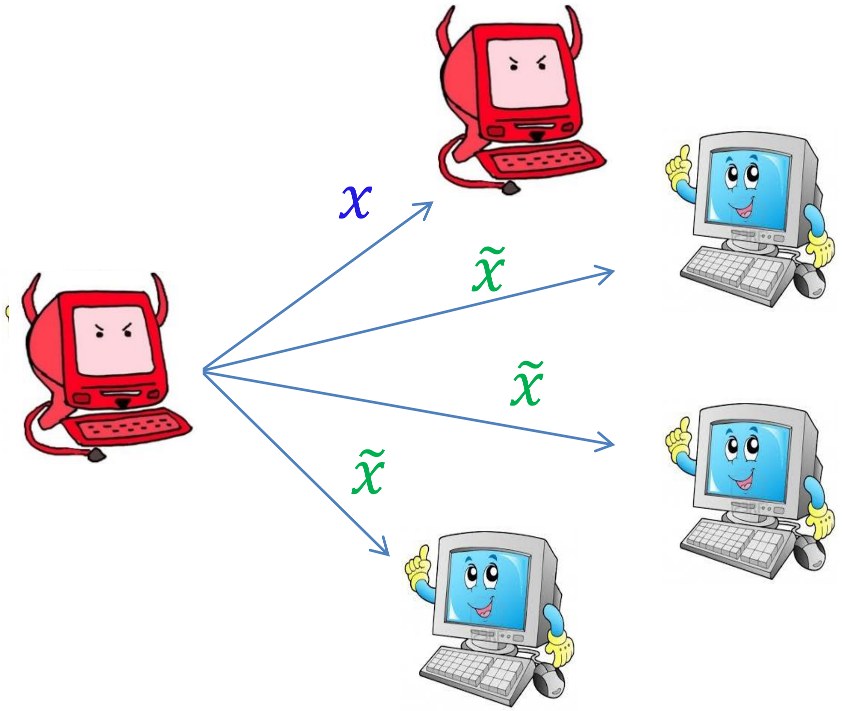
- The input x might be delivered first to a corrupt party (rushing adversary)
- If the adversary doesn't like x he can corrupt the sender and send $\tilde{x} \neq x$ instead (or crash)

What if we use a broadcast protocol?

(Almost) all known broadcast protocols follow this paradigm:

- Step 1: Sender sends its input x to every party
- Step 2: Parties try to establish agreement

All these protocols satisfy agreement and validity, even facing an adaptive adversary
Should they be considered adaptively secure?

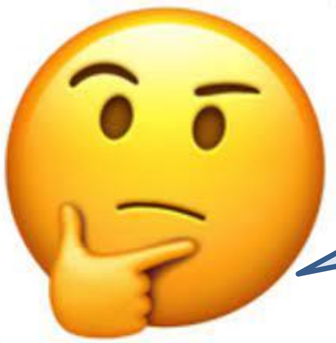


- The input x might be delivered first to a corrupt party (rushing adversary)
- If the adversary doesn't like x he can corrupt the sender and send $\tilde{x} \neq x$ instead (or crash)

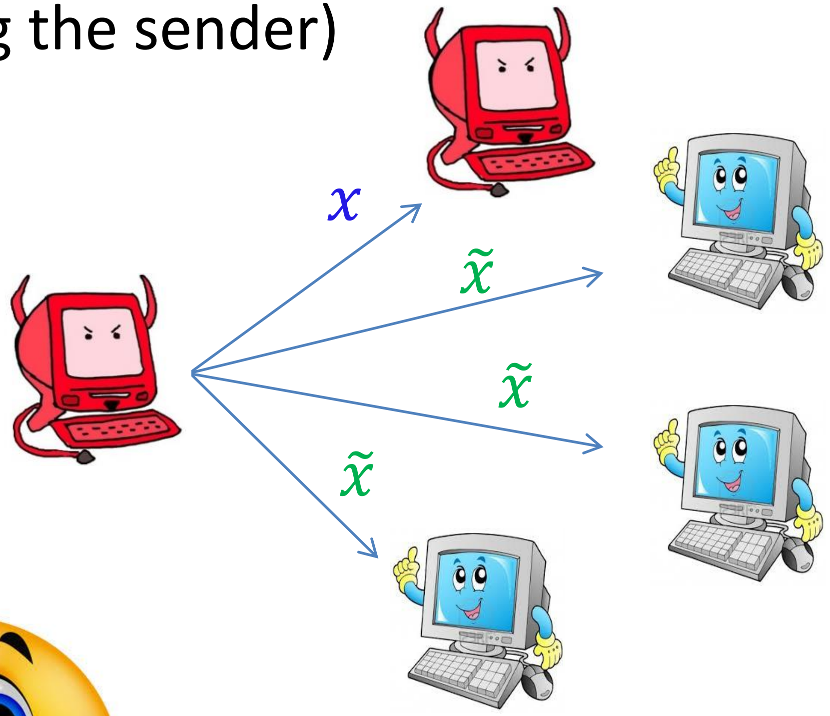
What if we use a broadcast protocol?

The adversary gets to:

1) Be the first to learn the sender input x
2) Decide whether to resume with x (without corrupting the sender) or corrupt the sender and change the input to $\tilde{x}$



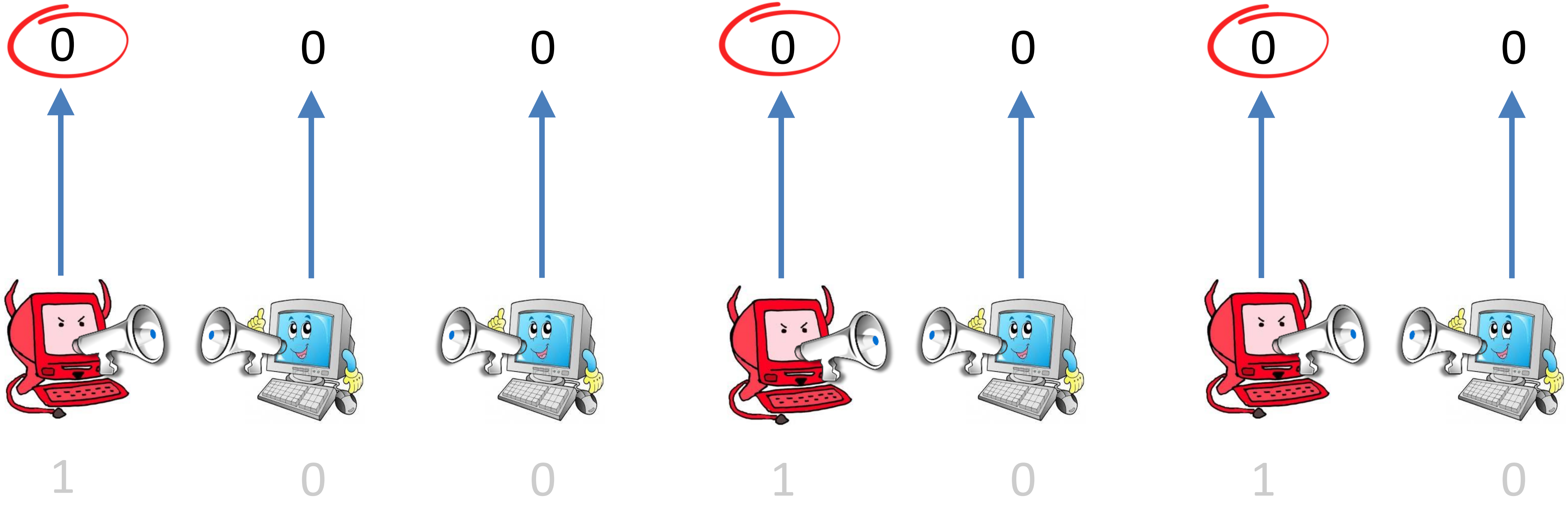
Should I be worried?
This attack seems to require
strong adversarial power



Think of message diffusion
mechanisms (à la Bitcoin,
Cardano, Algorand,…)

Case study: Adaptively Secure Broadcast

Problem: everybody broadcasts a bit; the adversary wants the output to be (as close as possible to) 0000…000

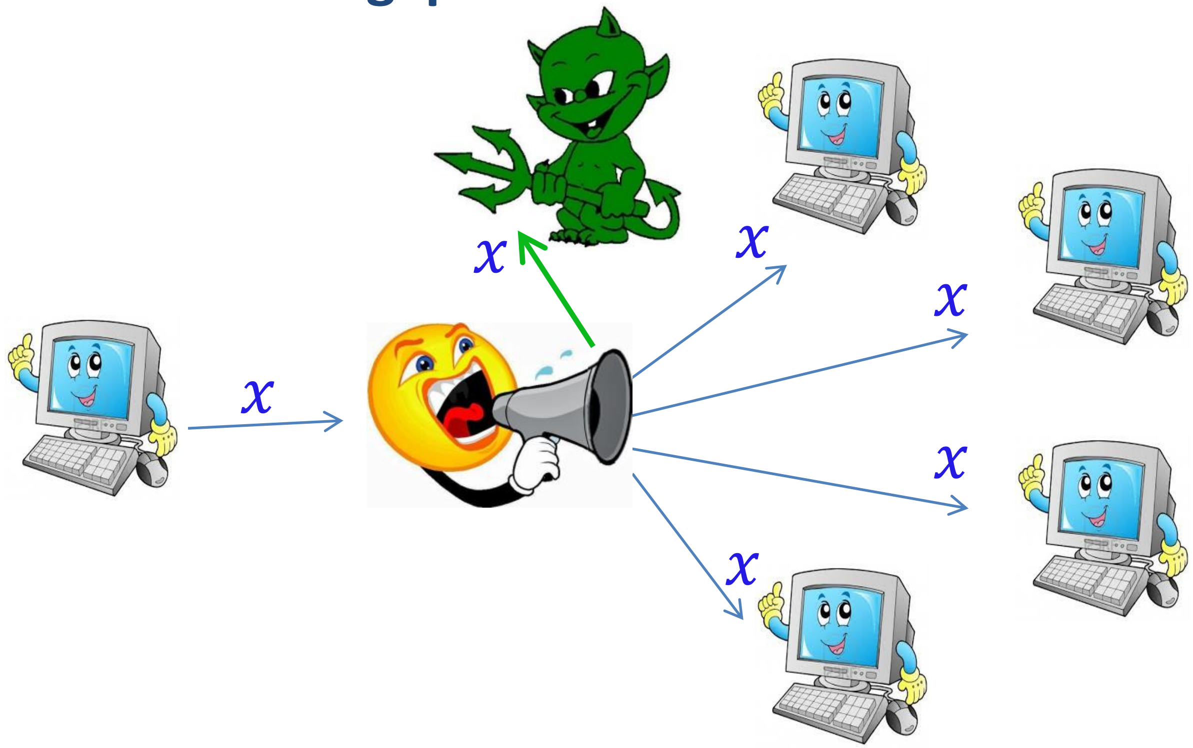


Adaptive adversary, 3 corruptions, standard broadcast protocol

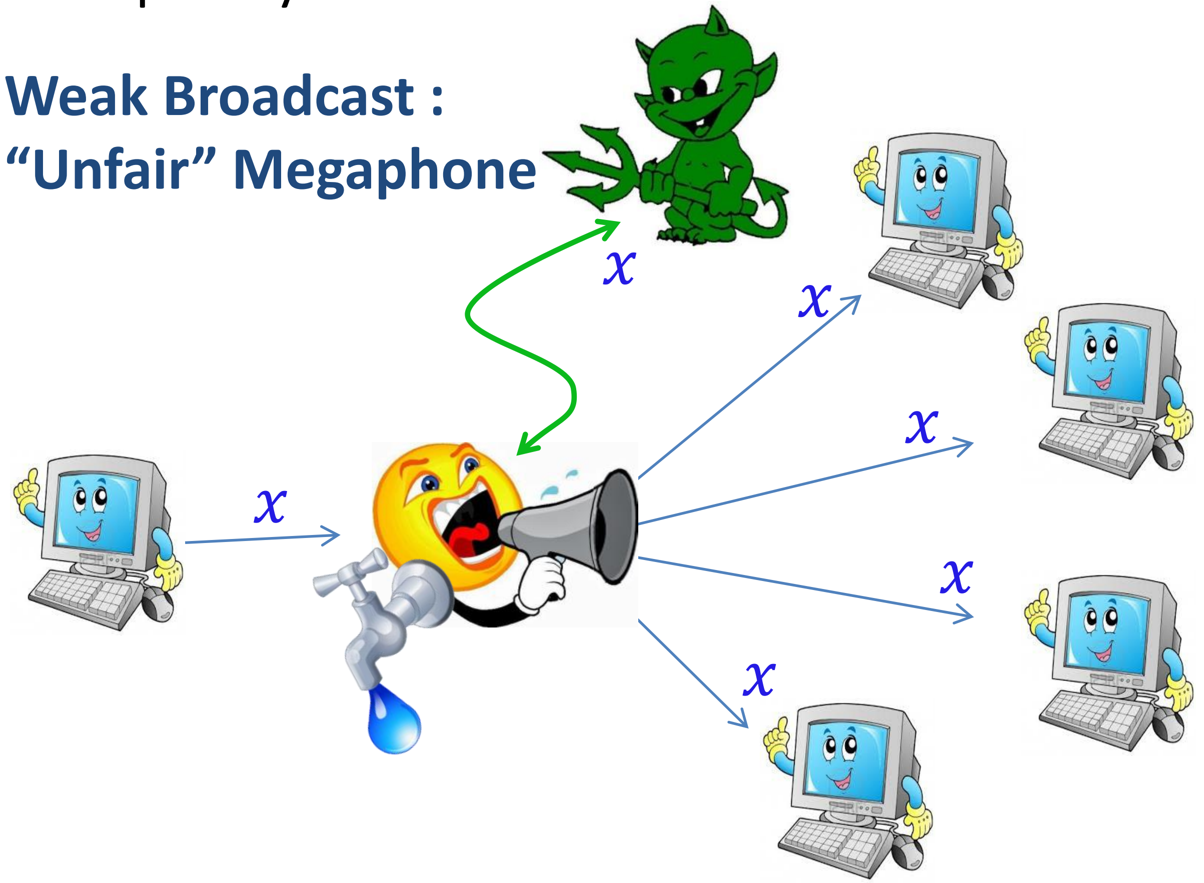
Simulation-based broadcast

Hirt and Zikas [EC'10]: simulation-based security of adaptively secure broadcast

Broadcast : Megaphone



Weak Broadcast : "Unfair" Megaphone



- Possible for $t < n/3$ without setup
- Possible for $t \leq n/2$ with PKI
- **Impossible for $t > n/2$ even with PKI**

Typical BC implement this with adaptive security:

- For $t < n/3$ without setup
- For $t < n$ with PKI

This is a very annoying impossibility…

Question: “This is an artifact of strong requirements of simulation-based (composable) security” [TCC’19,TCC’20a,TCC’20b]
Maybe using a weaker definition makes the impossibility go away?

Question: programmable random oracle can overcome many impossibilities regrading adaptive corruptions (e.g., Non-Committing Encryption)
Can we use RO to overcome also this impossibility?

Question: Time-Lock Puzzles (TLPs) hide information from rushing adversaries
Can we use TLPs to overcome also this impossibility?

Main Results

- **This is not an artifact of simulation-based security!**
- A new property for adaptively secure broadcast (**corruption-fairness**)
- Characterization of feasibility (for $t > n/2$)

	Property-based	Simulation-based
PKI	✗ $^{(*)}$	✗ [HZ'10]
PKI + RO	✗ $^{(*)}$	✗
PKI + TLP	✓	✗
PKI + TLP + RO	✓	✓

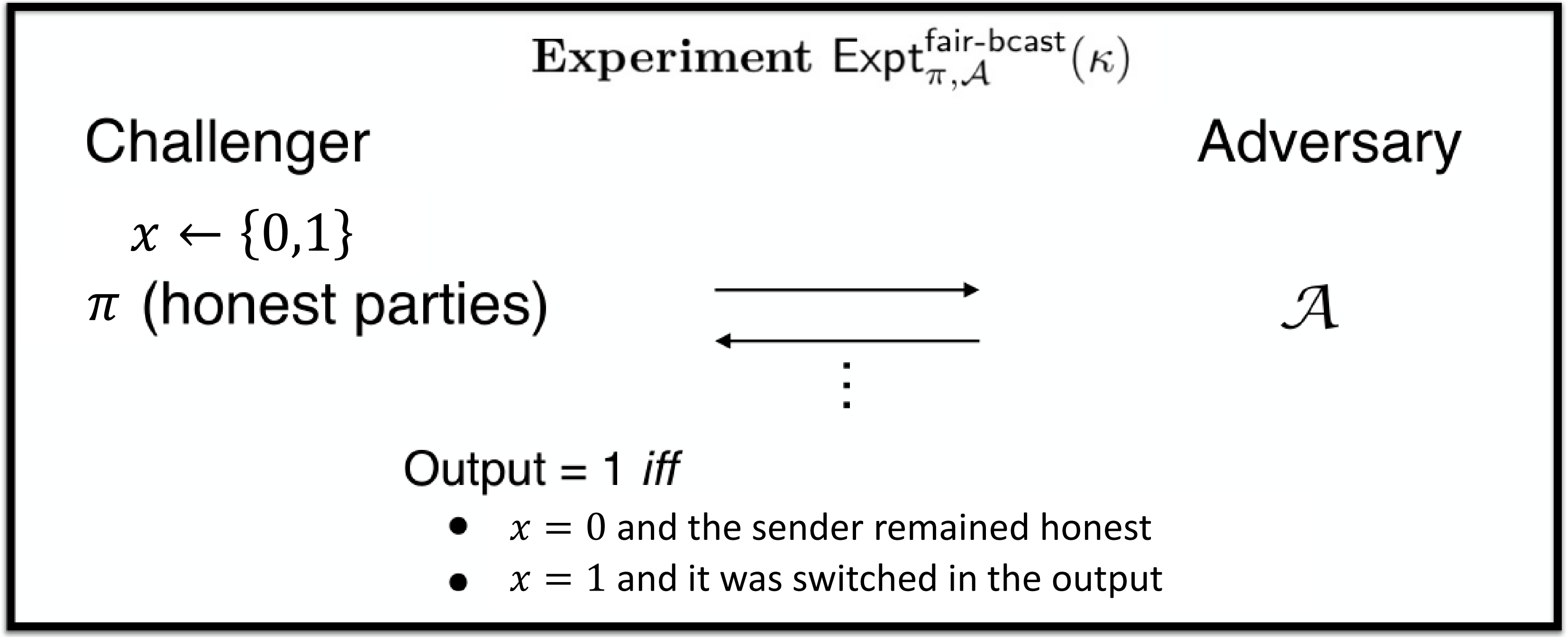
$(*)$ for a large class of broadcast protocols

- First (limited) composition theorem for resource-restricted adversaries

Corruption-Fairness

Informally: the adversary should not be able to:

- First learn the sender's input
- Based on the input value, corrupt the sender and affect honest parties' output



$$\pi \text{ is corruption-fair : } \Pr\left[\mathsf{Expt}_{\pi,\mathcal{A}}^{\mathsf{fair\text{-}bcast}}(\kappa) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\kappa)$$

Adaptively Secure Broadcast: Property-based

A broadcast protocol with sender S is considered adaptively secure if it satisfies the following properties:

- Agreement
- Validity
- Corruption-Fairness

Lemma (sanity check): this definition is implied by the simulation-based ("megaphone") definition

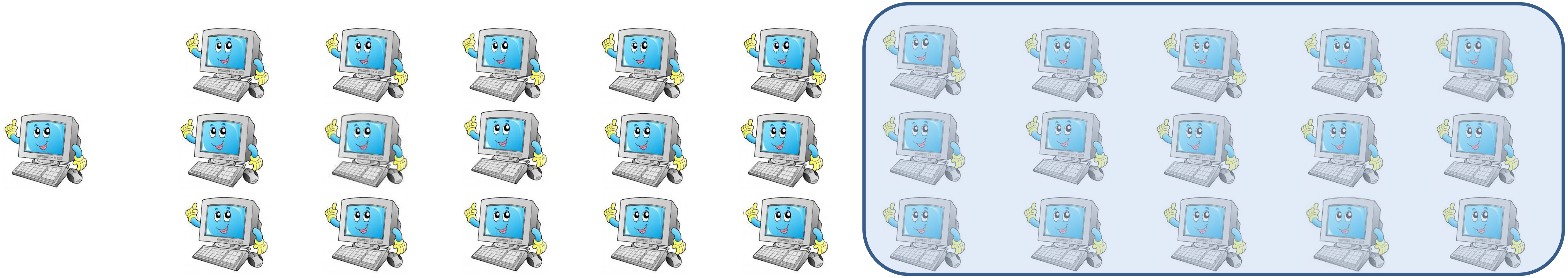
Impossibility of Property-based Broadcast

Protocol class Π^*: $\exists$ a round r^* and a set C^* of size $n/2-1$ such that

- Until round r^* no set of size $n/2-1$ (excluding the sender) knows the input x with certainty (i.e., if everyone else crash they will make a noticeable error)
- At round r^* parties in C^* know x (i.e., output x with overwhelming probability)

Uncertainty for any $n/2-1$ parties | r^* | C^* knows x | rounds

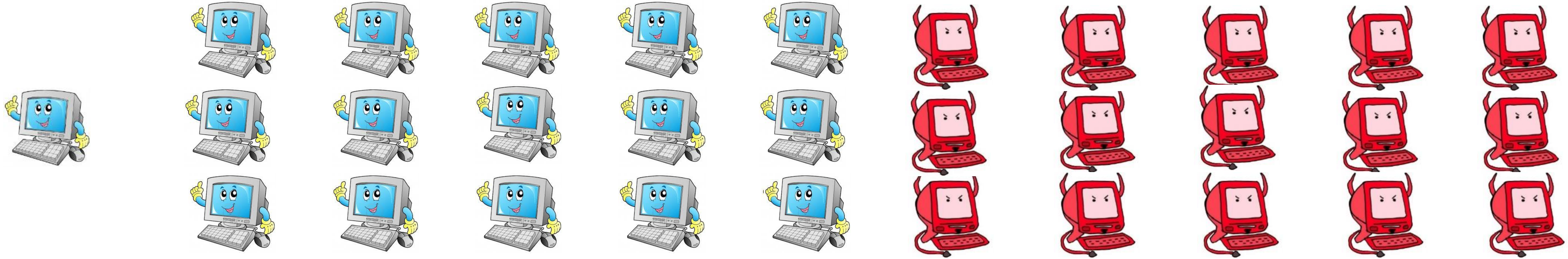
All broadcast protocols are in Π^* (with $r^*=1$)



Impossibility of Property-based Broadcast

Theorem 1: No protocol in Π^* is adaptively secure (**property-based**) against $> n/2$ corruptions

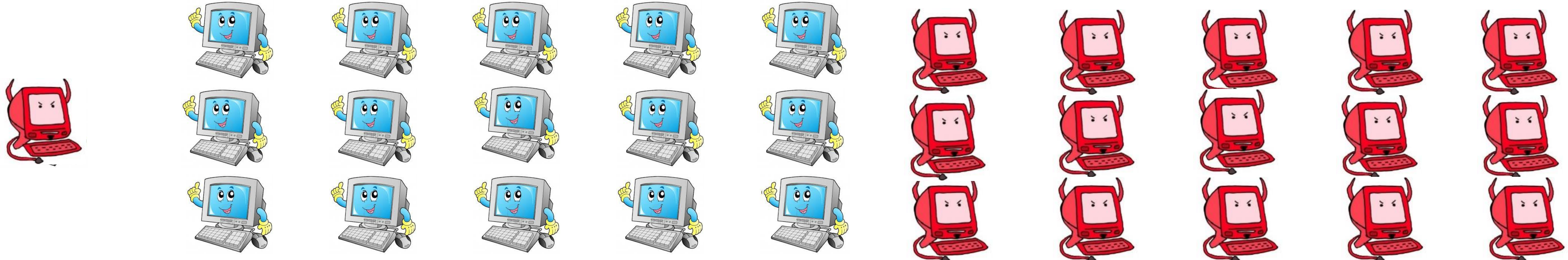
- The rushing adversary corrupts C^*
- At round r^* the adversary can learn the value x
  - If $x = 0$, the adversary lets the protocol complete



Impossibility of Property-based Broadcast

Theorem 1: No protocol in Π^* is adaptively secure (**property-based**) against $> n/2$ corruptions

- The rushing adversary corrupts C^*
- At round r^* the adversary can learn the value x
  - If $x = 0$, the adversary lets the protocol complete
  - If $x = 1$, the adversary crashes parties in C^* and the sender, **before** sending their round r^* messages



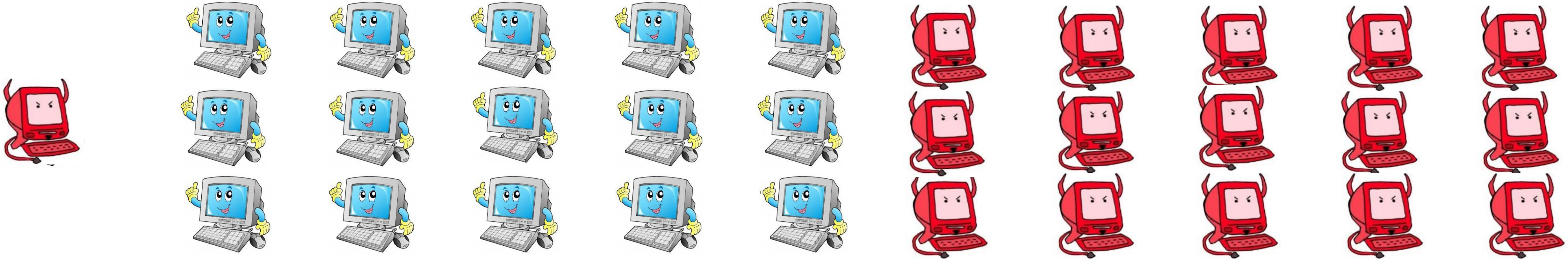
Impossibility of Property-based Broadcast

Theorem 1: No protocol in Π^* is adapti
against $> n/2$ corruptions

$\mathcal{A}$ **corrupts the sender with negligible probability**

- The rushing adversary corrupts C^*
- At round r^* the adversary can learn the value x
  - ➢ If $x = 0$, the adversary lets the protocol complete
  - ➢ If $x = 1$, the adversary crashes parties in C^* and the sender, **before** sending their round r^* messages

$\mathcal{A}$ **switches from 1 to 0 with noticeable probability**



Overcoming the impossibility?

- What if C^* has all the information to learn x in round r^*, but cannot access it until round $r^* + 1$ begins?
- In this case $\mathcal{A}$ doesn't know whether to corrupt the sender or not
- Intuitively, TLPs do exactly that
  - The sender can put the message in a TLP
  - Everyone who work enough will get the message
  - Anyone who doesn't work enough sees gibberish

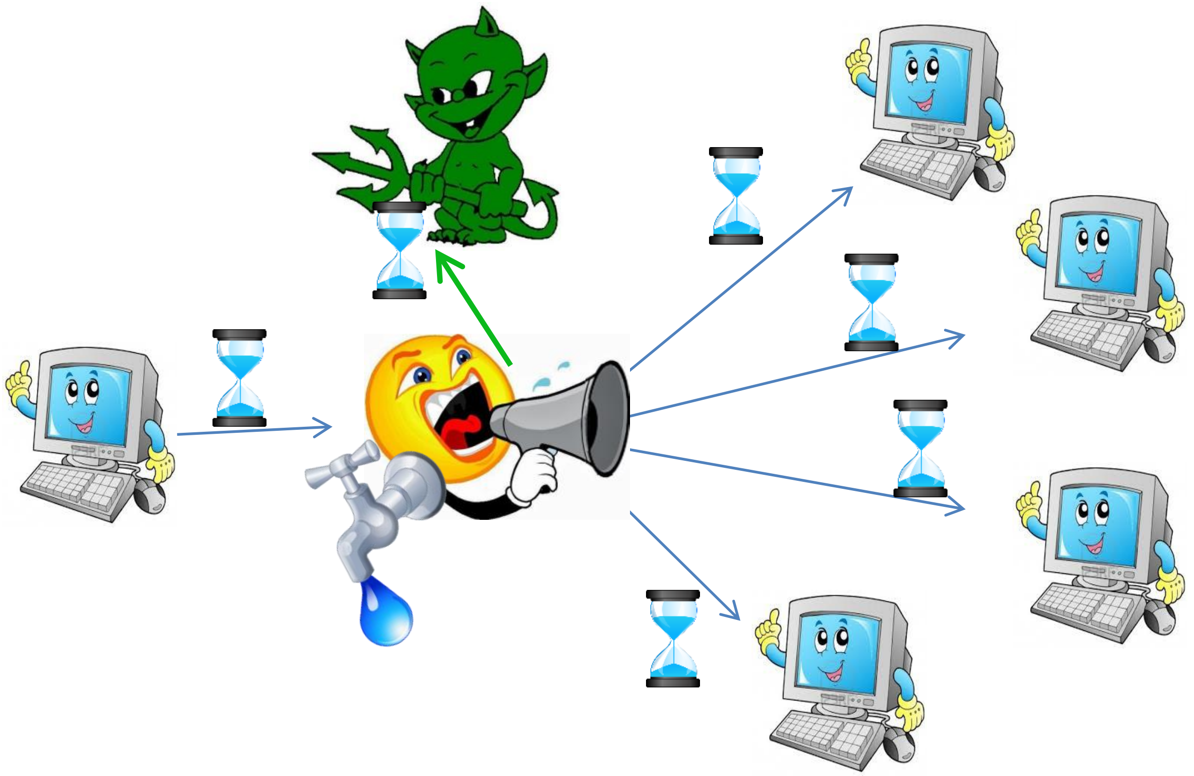
- Need to restrict the sequential speed of the adversary
  - A PPT adversary $\mathcal{A}$ is (R, T)-bounded if within R communication rounds, $\mathcal{A}$ can evaluate circuits of maximal depth T

Overcoming the impossibility?

Theorem 2: if corruption-unfair broadcast can be computed in R rounds, and the adversary is (R, T)-bounded, and TLPs exist, then there exists adaptively secure broadcast (**property-based**) for $t < n$ corruptions

Protocol:

1) Sender locks x in a TLP and sends using corruption-unfair broadcast
2) Once received, everyone works to open the TLP



Is the protocol simulation-based secure?

- When the sender is honest, Sim must simulate the puzzle
- But Sim doesn't know x at this point
  - If Sim asks the megaphone for x,
    then Sim gets stuck if $\mathcal{A}$ asks to corrupt the sender and change its input
  - If Sim doesn't ask the megaphone and commits to an arbitrary bit,
    then Sim gets stuck w.p. 1/2 if $\mathcal{A}$ lets the protocol complete without corrupting the sender

Theorem 3: No broadcast protocol is adaptively secure (**simulation-based**) against $> n/2$ corruptions, even assuming TLPs

Overcoming the impossibility?

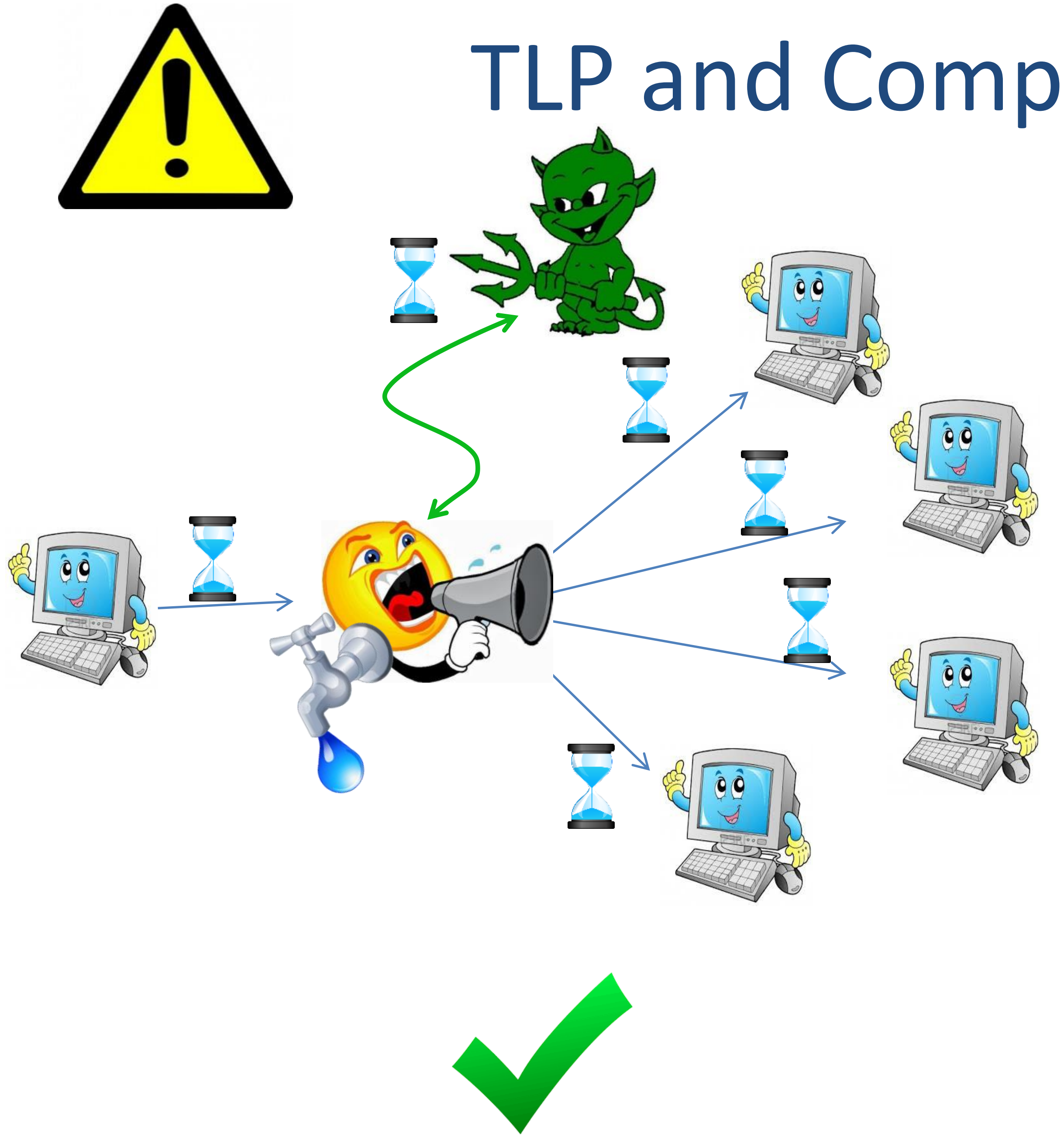
- The simulator got stuck because TLPs are committing
- Is it possible to make a TLP non-committing?
- Yes! In the programmable random oracle model

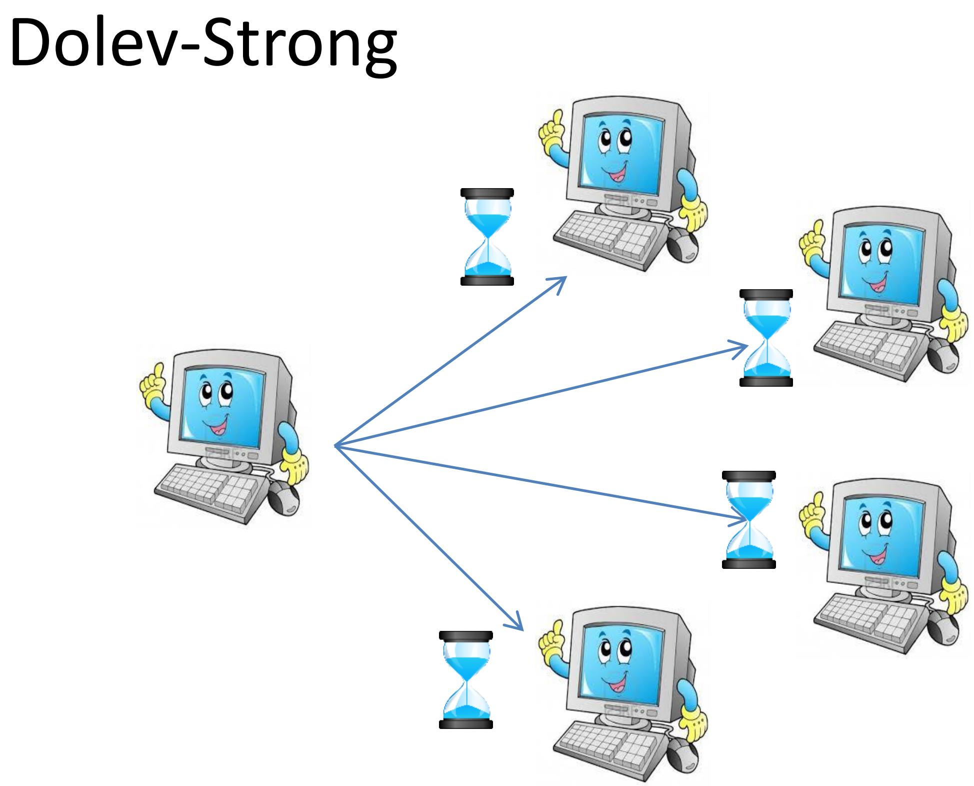
Protocol:

1) Sender locks r in a TLP and sends with $H(r) \oplus x$ using corruption-unfair broadcast
2) Once received, everyone works to open the TLP and recover x

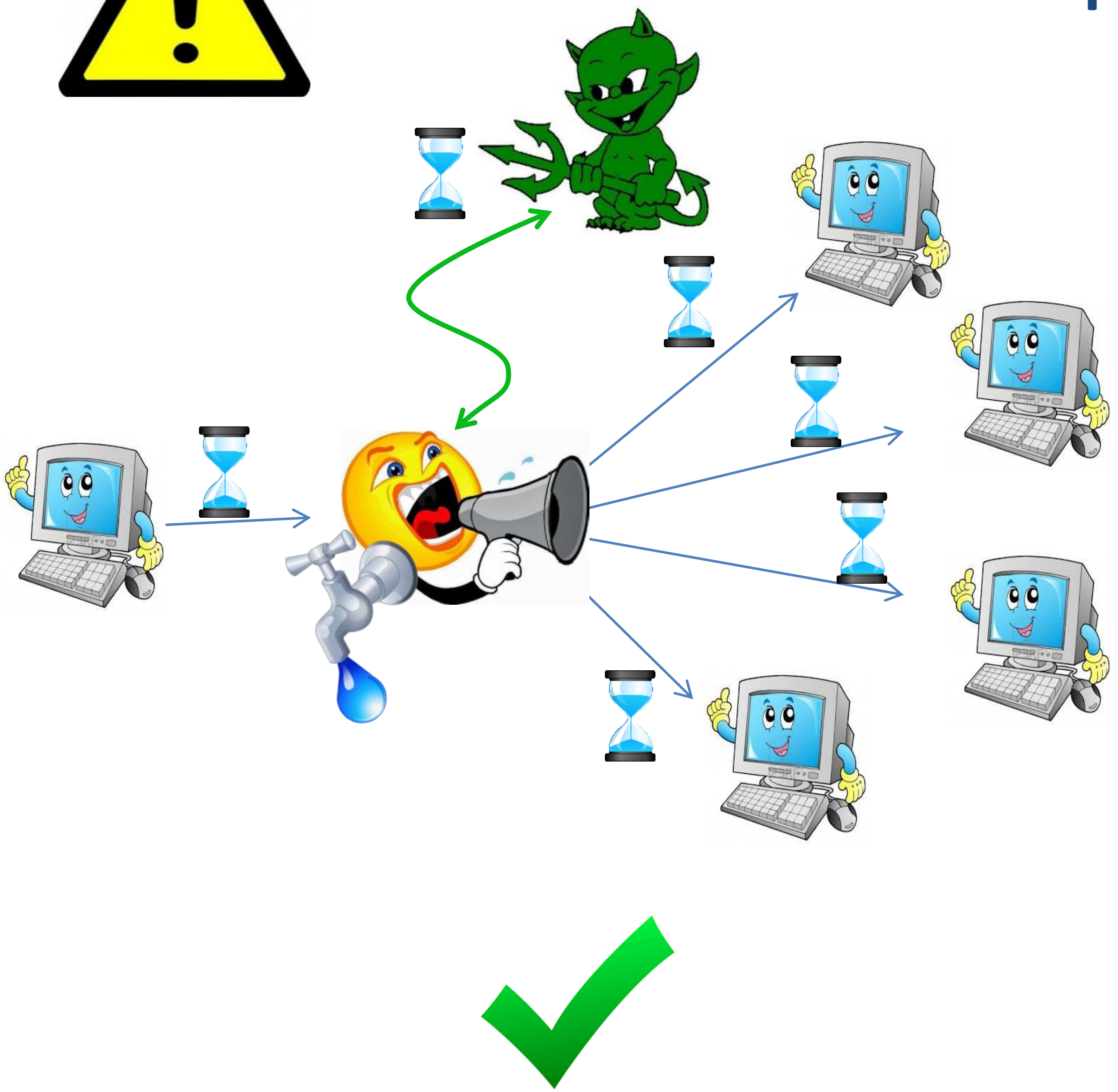
Theorem 4: if corruption-unfair broadcast can be computed in R rounds, and the adversary is (R, T)-bounded, and TLPs exist, then there exists adaptively secure broadcast (**simulation-based**) for $t < n$ corruptions in the programmable ROM

TLP and Composition

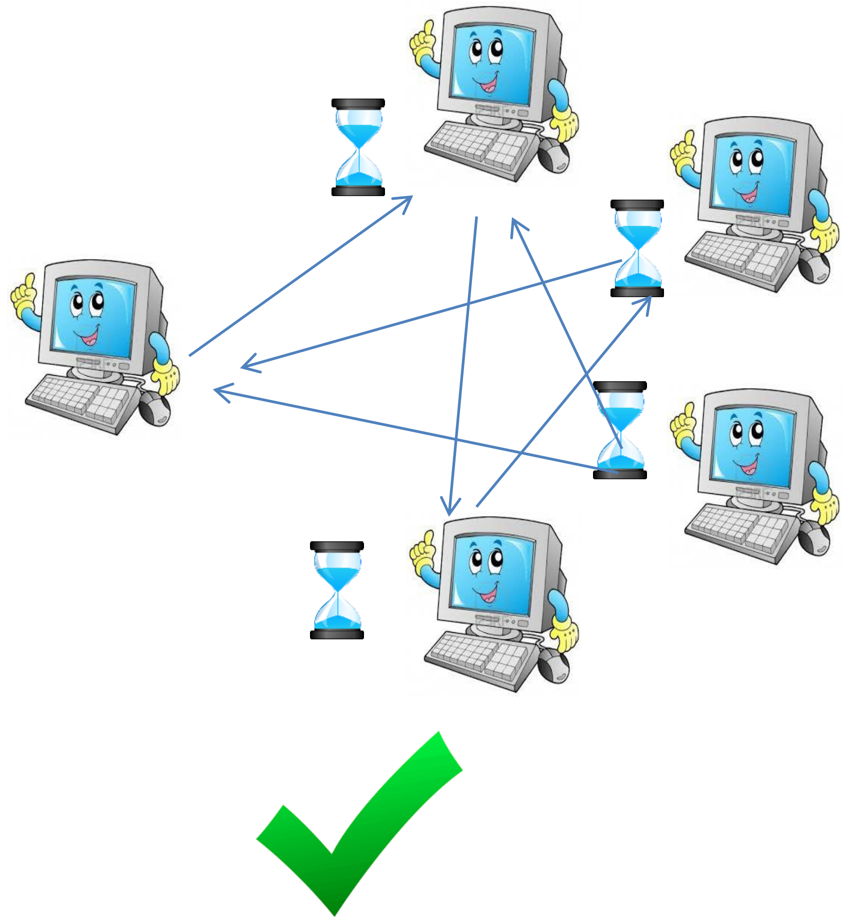




TLP and Composition



Dolev-Strong



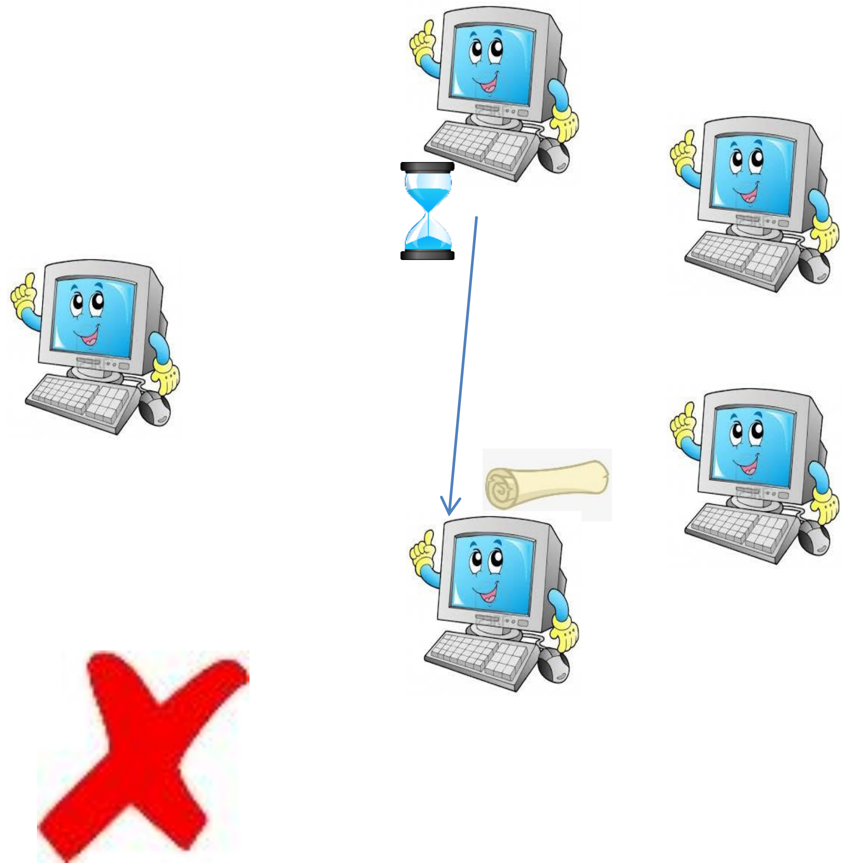
TLP and Composition

Adjusted Dolev Strong:

- Parties run Dolev-Strong
- During the protocol:
  - ➢ P_i generates a TLP and sends to P_j
  - ➢ P_j solves the returns answer to P_i

This is still a corruption-unfair broadcast!

But completely breaks our constructions



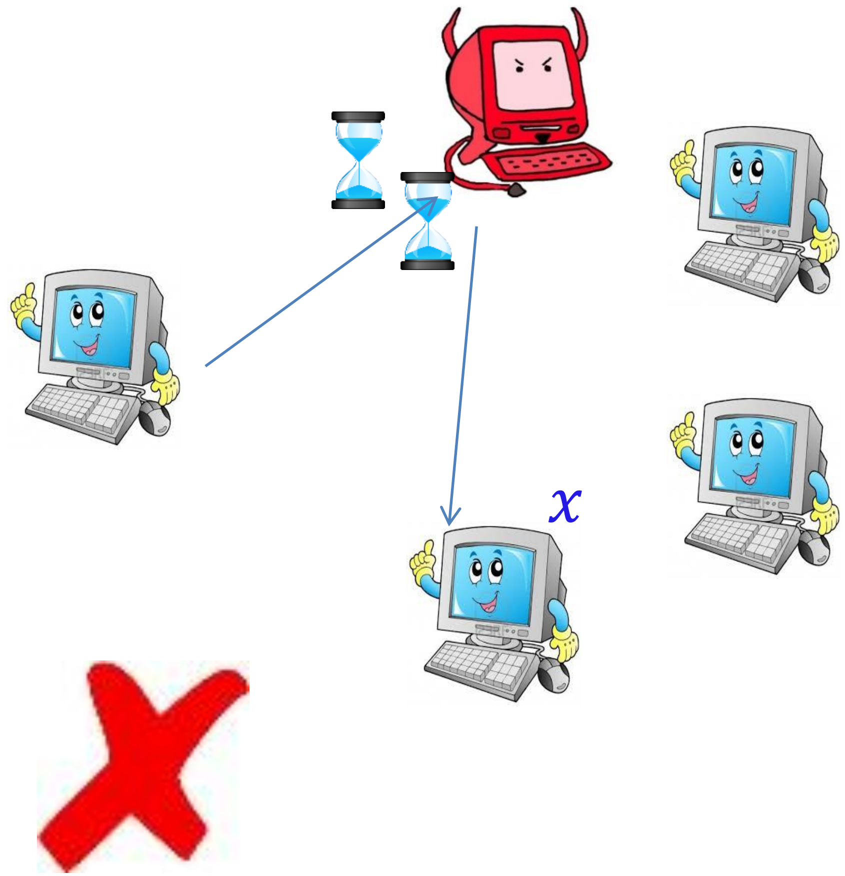
TLP and Composition

Adjusted Dolev Strong:

- Parties run Dolev-Strong
- During the protocol:
  - P_i generates a TLP and sends to P_j
  - P_j solves the returns answer to P_i

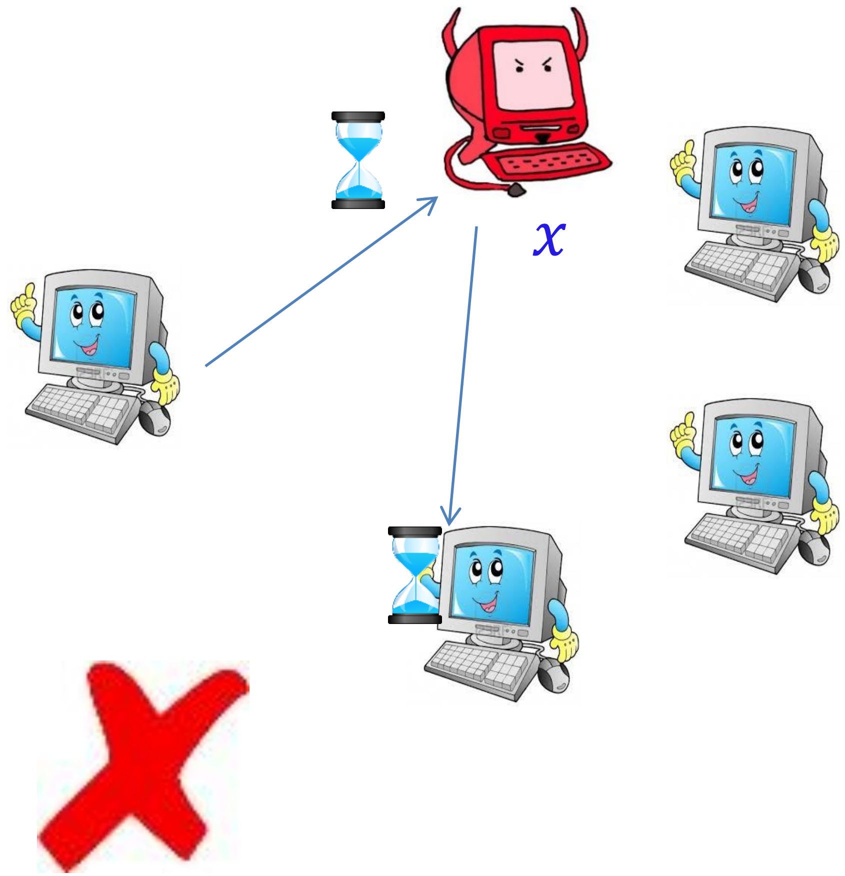
This is still a corruption-unfair broadcast!

But completely breaks our constructions



TLP and Composition

- Normally we restrict the sequential time of the adversary
- For composition we need to restrict honest parties as well
- Very tricky for simulation
- We prove the first (limited) composition theorem using a complexity-based definition of TLP



Summary

	Property-based	Simulation-based
PKI	✗ (*)	✗ [HZ'10]
PKI + RO	✗ (*)	✗
PKI + TLP	✓	✗
PKI + TLP + RO	✓	✓

Thank You